

AMENDMENTS TO THE CLAIMS

Pursuant to 37 C.F.R. § 1.121 the following listing of claims will replace all prior versions, and listings, of claims in the application.

1-35 cancel

36. (New) A method of monitoring data events occurring in a computer host application, the method comprising the steps of:

storing in a database a host application model, the model including a GuiFramework that models the host application interface controls that reflect the different states of the host application;

extracting the GuiFramework from the database and expanding the GuiFramework into an interlinked, indexed network structure in memory;

initializing data structures including a session structure;

intercepting, during execution of the host application, one or more operating system messages to obtain a plurality of information relating to a plurality of data events;

organizing the intercepted data events in the session structure; and

analyzing the data events to make automated inference of a user's interaction with the host application.

37. (New) The method of claim 36, further including the step of caching portions of the GuiFramework.

38. (New) The method of claim 36, wherein the data events include data events pertaining to session control information.

39. (New) The method of claim 36, further comprising the steps of:
monitoring all user actions with the host application;
creating a session monitor data set over a time period interval, wherein the session monitor data set is created by accumulating the monitored user actions into a monitoring database;
compiling the accumulated session monitor data set; and
analyzing the accumulated session monitor data set.

40. (New) The method of claim 39, wherein the time period interval is settable by the user.

41. (New) The method of claim 39, wherein the compiling step and the analyzing step occur at one of a scheduled time and by an invocation of the user.

42. (New) The method of claim 36, wherein the analyzing step further comprises the step of evaluating user inputs in real-time.

43. (New) The method of claim 36, further comprising the steps of:
separating actions involving tool bars and controls executing a single action function; and

46. (New) The method of claim 45, wherein the operating system messages include data events pertaining to session control information.

47. (New) The method of claim 45, wherein the interface device includes a mouse roller ball, a mouse button, or a keyboard.

48. (New) The method of claim 45, wherein the GuiFramework describes at least an interface hierarchy, and the searching step further comprises the step of searching for continuities of events that pass through consecutive portions of the interface hierarchy.

49. (New) The method of claim 45 further comprising the steps of:

providing an initial index that points to potential sequences detected during the searching step;

removing extraneous pointer device movements from the accumulated action structures; and

identifying a sequence that is a series of actions that accesses or navigates a frequent host application path.

50. (New) The method of claim 49, further including the step of the user inputting optional directives that assist later analysis.

51. (New) The method of claim 45, wherein the GuiFramework models navigational states of the host application.

52. (New) A method of monitoring host application events to locate frequent user sequences, comprising the steps of:

locating action events having the highest interface level position;

inspecting action structures for continuous property values representing ordinal positions with the host application;

identifying start and end points for the continuous action structures; and

generating reports and statistics relating to frequency of use and manner of use of interface objects accessed by the continuous action structures.

53. (New) A method of diagnosing operation of a host application operating under a computer operating system, the method comprising the steps of:

injecting a hook component and subclassing components into the host application, where the host application is supported by a container provided by the computer operating system;

intercepting events transmitted from the container or the operating system prior to the events reaching the host application;

interpreting the intercepted events into higher level logical events;

processing the higher level logical events to identify a sequence of events capable of being replicated; and

replicating the operation of the host application using the replicated sequences;

wherein the replicating step results in the diagnosis and analysis of the host application operation.